**Библиотека переводных публикаций
по функциональному программированию**

Ерон Фоккер

Систематическое конструирование однокомбинаторного базиса для λ-термов

Jeroen Fokker. The systematic construction of a one-combinator basis for Lambda-terms. — 1992

Проект журнала «Практика функционального программирования»

Этот перевод и другие материалы проекта «Библиотечка ПФП» доступны на fprog.ru/lib.

Библиотека переводных публикаций по функциональному программированию

Перевод: Роман Душкин

Ревизия: 3022 (2010-11-29)

Сайт проекта: http://fprog.ru/

Систематическое конструирование однокомбинаторного базиса для λ-термов

Ерон Фоккер
Кафедра компьютерной науки,
Университет Утрехта,
Нидерланды

1992

Аннотация

В этой статье описывается простое замкнутое λ-выражение, при помощи которого можно выразить все прочие λ-выражения. Его построение осуществляется систематическим образом. Полученное λ-выражение является более простым, чем известные из литературы однокомбинаторные базисы.[1]

Ключевые слова: Базис **S**, **K**, **I**, систематическое построение, простота в качестве цели.

[1]Запросы и прочую корреспонденцию отправлять Ерону Фоккеру (jeroen@cs.uu.nl), Кафедра компьютерной науки, Университет Утрехта, Нидерланды, P. O. Box 80.089, 3508 TB Utrecht, The Netherlands.

1. Комбинаторные базисы

Комбинатором называется замкнутый λ-терм, то есть выражение, состоящее из λ-абстракций и применений (аппликаций), не содержащее свободных переменных. Например, вот как выглядят несколько обычно используемых комбинаторов:

- $\mathbf{S} = \lambda f g x.fx(gx)$
- $\mathbf{K} = \lambda xy.x$
- $\mathbf{I} = \lambda x.x$

В этой статье строчные буквы будут использоваться для обозначения переменных в языке λ-исчисления, а заглавные буквы будут использоваться для обозначения λ-термов. Зафиксированные (именованные) выражения обозначаются прямым шрифтом. Применение является левоассоциативной операцией, поэтому выражение ABC обозначает $(AB)C$. Абстракция распространяется на всё выражение после символа точки (.), поэтому выражение $\lambda x.AB$ обозначает $\lambda x.(AB)$. Множественные абстракции записываются сокращённо, поэтому выражение $\lambda xy.A$ обозначает $\lambda x.\lambda y.A$.

Комбинаторы имеют важнейшее значение при реализации функциональных языков программирования [3]. Функциональные программы могут быть выражены через набор последовательных применений комбинаторов друг к другу, поэтому отсутствует необходимость в λ-абстракции. Набор комбинаторов, с помощью которых посредством применения (аппликации) могут быть построены все замкнутые λ-термы (с точностью до α-, β- и η-конверсии), называется базисом. Х. Карри [2] показал, что конечный базис существует:

Теорема 1. *Набор комбинаторов* **S**, **K**, **I** *является базисом.*

Доказательство. Любое заданное λ-выражение выражается через указанные комбинаторы при помощи последовательного применения следующих правил, убирающих абстракции:

1) $\lambda x.x \quad \Rightarrow \quad \mathbf{I}$
2) $\lambda x.A \quad \Rightarrow \quad \mathbf{K}A$, если $x \notin Free(A)$
3) $\lambda x.AB \quad \Rightarrow \quad \mathbf{S}(\lambda x.A)(\lambda x.B)$

Эти правила покрывают собой все возможные варианты, так как тело λ-абстракции является либо связанной переменной (правило 1), либо другой переменной (правило 2), либо применением (правило 3), либо другой абстракцией, которая сама должна быть убрана в первую очередь. Процесс перевода в указанный базис останавливается, как только он выдал набор применений меньших термов. Правило 1 являет собой определение комбинатора **I**, правило 2 корректно по определению комбинатора **K**:

$$
\begin{array}{lcl}
\mathbf{K}A & = & \{\text{ Определение } \mathbf{K} \} \\
(\lambda xy.x)A & = & \{\ \beta\text{-редукция } \} \\
\lambda y.A & = & \{\ \alpha\text{-конверсия, } x \notin Free(A) \} \\
\lambda x.A & &
\end{array}
$$

Правило 3 соответственно верно по определению комбинатора **S**:

$$
\begin{array}{lcl}
\mathbf{S}(\lambda x.A)(\lambda x.B) & = & \{\text{ Определение } \mathbf{S} \} \\
(\lambda fgx.fx(gx))(\lambda x.A)(\lambda x.B) & = & \{\ \beta\text{-редукция (дважды) } \} \\
\lambda x.(\lambda x.A)x((\lambda x.B)x) & = & \{\ \eta\text{-конверсия (дважды) } \} \\
\lambda x.AB & &
\end{array}
$$

□

М. Шейнфинкель [4][2] заметил, что комбинатор **I** может быть выражен через комбинаторы **S** и **K**:

Теорема 2. *Набор комбинаторов* **S**, **K** *является базисом.*

Доказательство. По теореме 1 набор комбинаторов **S**, **K**, **I** является базисом. Все появления комбинатора **I** необходимо заменить на выражение **SK**A, где A — произвольный терм, к примеру — **K**. Доказательство выглядит следующим образом:

$$
\begin{array}{lcl}
\mathbf{SK}A & = & \{\text{ Определение } \mathbf{S} \} \\
(\lambda fgx.fx(gx))\mathbf{K}A & = & \{\ \beta\text{-редукция (дважды) } \} \\
\lambda x.\mathbf{K}x(Ax) & = & \{\text{ Определение } \mathbf{K} \} \\
\lambda x.(\lambda xy.x)x(Ax) & = & \{\ \beta\text{-редукция (дважды) } \} \\
\lambda x.x & &
\end{array}
$$

□

Центральной точкой рассмотрения этой статьи является поиск некоторого комбинатора **X**, составляющего базис.

2. Однокомбинаторный базис

Далее будет сконструирован комбинаторный базис, состоящий из одного комбинатора. Важность такого базиса является больше теоретической. Применяющиеся на практике базисы обычно имеют больший, а не меньший, размер, чем базис **S**, **K**, **I**. Приведённое построение является примером систематического вывода решения из спецификации задачи. В процессах трансформации программ обычно некоторые шаги являются типовыми и строго обусловленными

[2] В оригинальной статье нет ссылки на фундаментальную работу М. Шейнфинкеля — прим. перев.

контекстом, но некоторые шаги проводятся по наитию. В конструировании, приведённом ниже, определено три таких интуитивных решения — было требование минимизировать количество таких шагов. Простота являлась движущей силой в построении базиса для получения наиболее «естественного» результата.

Целью процесса является создание комбинатора **X**, при помощи которого можно получить комбинаторы **S** и **K** посредством операции применения. Затем по теореме 2 может быть построен любой замкнутый λ-терм. Ни один из комбинаторов **S** или **K** не являет собой однокомбинаторный базис, поэтому оба этих комбинатора должны быть построены из нескольких экземпляров комбинатора **X**.

Первым интуитивным шагом в построении однокомбинаторного базиса является предположение о том, что комбинаторы **S** и **K** являются применением, с левой стороны которого находится комбинатор **X**. Далее необходимо найти правые стороны этих применений:

- $\mathbf{X}A = \mathbf{K}$
- $\mathbf{X}B = \mathbf{S}$

Естественно, что термы A и B должны быть различными. Вторым интуитивным шагом будет предположение о том, что эти термы являются простейшими различными выражениями, которые можно получить из комбинатора **X**, а выражение для комбинатора **K** должно быть проще, чем такое для комбинатора **S**, то есть терм A должен быть проще терма B:

- $A = \mathbf{X}$
- $B = \mathbf{XX}$

Теперь выражение комбинаторов **S** и **K** выглядит следующим образом:

- $\mathbf{XX} = \mathbf{K}$
- $\mathbf{X}(\mathbf{XX}) = \mathbf{S}$

(М. Шейнфинкель уже давал такую спецификацию в своей работе [4], однако с изменёнными местами для термов A и B. Но он не предоставил закрытую форму для комбинатора **X**). Используя первое представленное выражение во втором, можно получить:

- $\mathbf{XX} = \mathbf{K}$
- $\mathbf{XK} = \mathbf{S}$

Так как комбинатор **X** применяется к функциям (**X** и **K**), то его форма должна быть такой: $\lambda f.M$. Теперь необходимо сконструировать тело выражения M вне определения f. Не очень ясно, где внутри выражения M должна использоваться переменная f. Но эта переменная являет собой функцию, поэтому ясно, что она должна к чему-то применяться. Так как функция f может быть выражена через **K**, а у этого комбинатора два аргумента, то можно попробовать также применить функцию f к двум аргументам. Поэтому:

$$\mathbf{X} = \lambda f.fPQ$$

Необходимо в этом уравнении найти выражения P и Q. Можно провести следующие вычисления:

S	=	{ Спецификация }
XK	=	{ Определение **X** }
KPQ	=	{ Определение **K** }
P		

Таким образом видно, что $P = \mathbf{S}$. Далее:

K	=	{ Спецификация }
XX	=	{ Определение **X** }
XPQ	=	{ Определение **X** }
$PPQQ$	=	{ $P = \mathbf{S}$ }
SSQQ	=	{ Определение **S** }
S$Q(QQ)$		

Для редукции комбинатора **S** на этом шаге требуется три аргумента. Можно использовать принцип экстенсиональности и предположить для любых A и B:

A	=	{ Определение **K** }
KAB	=	{ Вышеприведённое выражение }
S$Q(QQ)AB$	=	{ Определение **S** }
$QA(QQA)B$		

Последнее выражение полностью удовлетворяется формой $Q = \lambda xyz.x$. Этот вывод можно зафиксировать в следующей теореме:

Теорема 3. *Пусть* $\mathbf{X} = \lambda f.f\mathbf{S}(\lambda xyz.x)$, *тогда этот комбинатор составляет однокомбинаторный базис.*

Доказательство. По теореме 2 набор **S**, **K** является базисом. Все появления комбинатора **K** необходимо заменить на выражение **XX**, а появление комбинатора **S** на выражения **X**(**XX**). Правильность таких замен показана выше. □

Таблица 1. Однокомбинаторные базисы

Meredith	$\mathbf{X}$	$=$	$\lambda abcd.cd(a(\lambda x.d))$
	$\mathbf{U}$	$=$	$\mathbf{X}^3\mathbf{X}^2$
	$\mathbf{V}$	$=$	$\mathbf{X}^4(\mathbf{K}\mathbf{X}^4)$
	$\mathbf{K}$	$\Rightarrow$	$\mathbf{X}^4(\mathbf{X}^4\mathbf{U}\mathbf{X}^2)\mathbf{X}^2$
	$\mathbf{S}$	$\Rightarrow$	$\mathbf{V}(\mathbf{U}(\mathbf{X}^4(\mathbf{X}^4\mathbf{V}^2)(\mathbf{K}\mathbf{X}^4)))$
Böhm	$\mathbf{X}$	$=$	$\lambda f.f(f\mathbf{S}(\mathbf{K}^3\mathbf{I}))\mathbf{K}$
	$\mathbf{K}$	$\Rightarrow$	$\mathbf{X}\mathbf{X}$
	$\mathbf{S}$	$\Rightarrow$	$\mathbf{X}(\mathbf{X}\mathbf{X})$
Barendregt	$\mathbf{X}$	$=$	$\lambda f.f(f\mathbf{S}(\mathbf{K}\mathbf{K}))\mathbf{K}$
	$\mathbf{K}$	$\Rightarrow$	$\mathbf{X}\mathbf{X}\mathbf{X}$
	$\mathbf{S}$	$\Rightarrow$	$\mathbf{X}(\mathbf{X}\mathbf{X}\mathbf{X})$
Rosser	$\mathbf{X}$	$=$	$\lambda f.f\mathbf{K}\mathbf{S}\mathbf{K}$
	$\mathbf{K}$	$\Rightarrow$	$\mathbf{X}\mathbf{X}\mathbf{X}$
	$\mathbf{S}$	$\Rightarrow$	$\mathbf{X}(\mathbf{X}\mathbf{X})$
Fokker	$\mathbf{X}$	$=$	$\lambda f.f\mathbf{S}(\lambda xyz.x)$
	$\mathbf{K}$	$\Rightarrow$	$\mathbf{X}\mathbf{X}$
	$\mathbf{S}$	$\Rightarrow$	$\mathbf{X}(\mathbf{X}\mathbf{X})$

3. Сравнение с другими базисами

В [1] даны некоторые иные однокомбинаторные базисы. Первый был найден независимо Мередитом в 1963 году и Барендрегтом в 1971 году. Другие формы были найдены Бёмом и Россером. В таблице 1 приведены выражения из [1]. Проявления в определениях **X** комбинаторов **S**, **K**, **I** должно заменять на λ-термы, показанные в разделе 1.

Для того чтобы оценить простоту различных представленных решений, можно определить понятие *размера* как суммарное количество абстракций и аппликаций в заданном выражении. Тогда простота решения будет выражаться как размер представления комбинаторов **X**, **K** и **S**. В таблице 2 показаны эти размеры. Другим критерием простоты может быть количество шагов редукции по нормальной редукционной стратегии, которые требуются для достижения нормальной формы, а также сумма размеров промежуточных выражений в ходе проведения редукции. Эти величины также показаны в таблице 2. Решение, которое на систематической основе получено в этой статье, имеет минимальный размер и сравнимо с решением Россера по скорости нормализации.

Таблица 2. Простота базисов из таблицы 1

	размер			**K**		**S**	
	X	**K**	**S**	шагов	∑ размеров	шагов	∑ размеров
Meredith	74	152	380	41	3622	68	17681
Böhm	23	47	71	29	2509	63	7792
Barendregt	18	56	75	24	1803	53	5923
Rosser	14	44	44	8	262	11	316
Fokker	12	25	38	9	167	12	352

4. Иной способ решения

Решение Россера может быть вычислено примерно также, как это сделано в данной статье в разделе 2. Спецификация комбинаторов **S** и **K** в решении Россера такова (необходимо обратить внимание на дополнительный комбинатор **X** в первой строке):

- $\mathbf{XXX} = \mathbf{K}$
- $\mathbf{X}(\mathbf{XX}) = \mathbf{S}$

Достаточным условием для выражения **XXX** является выражение $\mathbf{XX} = \mathbf{KK}$ (и, поэтому, выражение $\mathbf{XX}X = \mathbf{K}$). Как и в основном решении можно подставить это достаточное условие в определение комбинатора **S**:

- $\mathbf{XX} = \mathbf{KK}$
- $\mathbf{X}(\mathbf{KK}) = \mathbf{S}$

Теперь необходимо отметить, что $\mathbf{KK}ABC = B$, поэтому выражение **KK** требует три аргумента. Таким образом, в этом конструировании можно предположить, что комбинатор **X** имеет форму:

$$\mathbf{X} = \lambda f.fPQR$$

Как и ранее теперь можно вычислить выражение $\mathbf{S} = \mathbf{X}(\mathbf{KK}) = \mathbf{KK}PQR = Q$, а также выражение $\mathbf{KK} = \mathbf{XX} = \mathbf{X}PQR = PPQRQR = PP\mathbf{S}R\mathbf{S}R$, так что:

- $Q = \mathbf{S}$
- $PP\mathbf{S}R\mathbf{S}R = \mathbf{KK}$

Здесь представлено одно уравнение с двумя неизвестными (P и R). У него много решений, некоторые из которых можно найти с лёгкостью. Например:

- $P = \lambda abcde.e$ и $R = \mathbf{KK}$
- $P = \lambda abcde.c$ и $R = \mathbf{KK}$
- $P = \lambda abcde.ce$ и $R = \mathbf{K}$
- $P = \mathbf{K}$ и $R = \mathbf{K}$

И только последнее решение не так очевидно, как все остальные. Оно показывается при помощи выражения $\mathbf{KKSKSK} = \mathbf{KKSK} = \mathbf{KK}$. Из этого решения следует выражение Россера для комбинатора $\mathbf{X}$: $\lambda f.f\mathbf{KSK}$. Именно степень свободы в выборе термов P и Q для этого решения мотивировала автора статьи начать поиск более простого решения.

Список литературы

[1] *Barendregt H. P.* The Lambda Calculus: Its Syntax and Semantics. — Revised edition. — North Holland, 1984. — Vol. 103.

[2] *Curry H. B., Feys R.* Combinatory Logic. — Third printing edition. — North-Holland Publishing Company, Amsterdam, 1974. — Vol. I.

[3] *Peyton Jones S. L.* The Implementation of Functional Programming Languages. Series in Computer Science. — Prentice-Hall International, 1987.

[4] *Schönfinkel M.* Über die bausteine der mathematischen logik // *Mathematische Annalen*. — 1924. — Vol. 92. — P. 305.